****

**Laboratório de Computadores**

**2018/2019**

Pen-Goo

**Imagem fixe**

**(ou o logotipo)**

Grupo T4G10:

DANIEL

MOAS

Índice

[1.Instrucões de Utilização 3](#_Toc534480905)

[1.1.Ecrã Inicial 3](#_Toc534480906)

[1.2.Menu Principal 3](#_Toc534480907)

[1.3 Draw 5](#_Toc534480908)

[1.4 Play (Pen-Goo) 7](#_Toc534480909)

[1.5 Draw Together 9](#_Toc534480910)

[1.6 Rainbow Snake 9](#_Toc534480911)

[1.7 Flappy Rainbow 10](#_Toc534480912)

[2. Estado do Projeto 11](#_Toc534480913)

[3. Organição / Estrutura de código 13](#_Toc534480914)

[Call Graph 18](#_Toc534480915)

[4.Detalhes de implementacão 19](#_Toc534480916)

[5.Conclusao 20](#_Toc534480917)

# 1.Instrucões de Utilização

## 1.1.Ecrã Inicial

Ao iniciar o jogo, é mostrado o ecrã inicial. Neste é possivel ver as horas / data atual e avançar para o menu principal , pressionando a tecla “Enter”.

Figura - Ecrã inicial

## 1.2.Menu Principal

A partir do menu principal , é possivel aceder a todos os modos de jogo e respetivas intrucões , assim como o botão de saída.

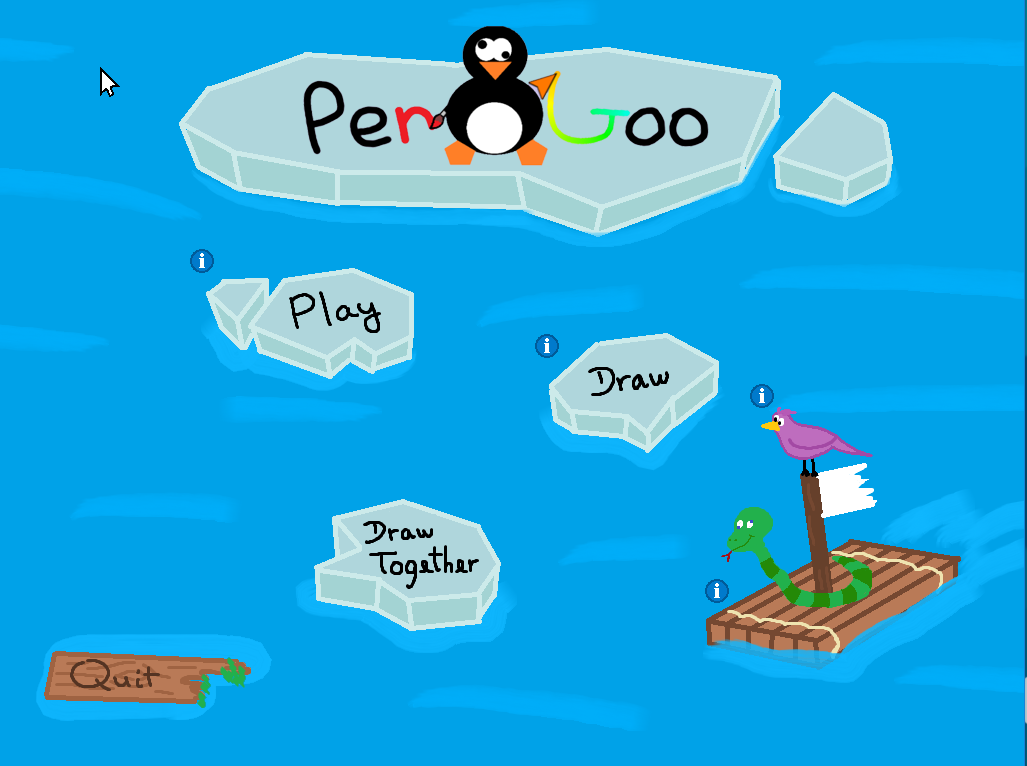


Figura - Menu Principal

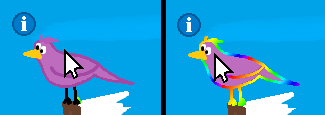
Para navegar neste menu , é utilizado o rato. Passar com o rato por cima de um dos butões faz com que esse fique “highlighted” , e clicar com o lado esquerdo do rato seleciona a opção correspondente. //acho que falta falar aqui das colisoes do rato com os butoes

Figura - "hovering" com o rato

Os modos de jogo disponíveis são: Play , Draw , Draw Together , Rainbow Snake e Flappy Rainbow.



Figura - Instrucões do modo Flappy Rainbow

## 1.3 Draw

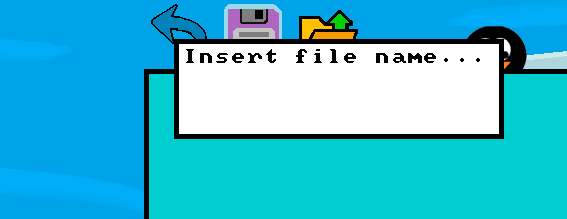
Modo singleplayer , onde o utilizador pode criar os seus próprios desenhos.

Figura 6 - Inserir nome para guardar ficheiro

Figura 5 - Modo Draw

Para além de começar um desenho em branco , é possivel fazer load a ficheiros em formato bitmap e guardar as criacões.

Para criar o desenho , o utilizador tem acesso a multiplas ferramentas de desenho:

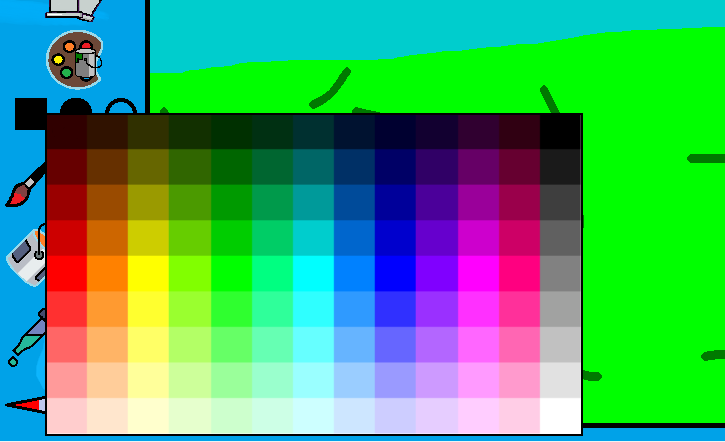
* **Slider de grossura** – arrastar o cursor para aumentar ou diminuir a grossura do pincel.

Figura 7 - Palete de cores

* **Palete de cores –** abre uma palete de cores , de onde se pode escolher a cor principal (butão esquerdo do rato) e secundária (butão direito do rato)
* **Pincel –** ferramenta de desenho principal. Cor principal (botão esquerdo do rato) e secundária (botão direito do rato)
* **Arco-Íris –** Pincel com gradiente arco-íris
* **Balde –** Preenche uma area com a mesma cor
* **Borracha –** Usar o rato para apagar. Botão direito utiliza a cor principal e botão esquerdo a secundária.
* **Color-Picker –** Clicando em qualquer área do ecrã consegue-se obter qualquer cor.
* **Lixo –** Apaga o desenho.
* **Formas –** Permite desenhar formas com dois cliques do rato. Retângulo: Dois cliques representam dois cantos opostos. Círculo/Circunferência: Primeiro clique para o centro, segundo para o raio.

Para a maioria das ferramentas é possivel utilizar Cor principal (botão esquerdo do rato) e secundária (botão direito do rato), exceto para a borracha, na qual acontece o inverso.

Para além das ferramentas de desenho é possivel:

* Ver cor principal e secundária atual utilizando os guaches de cor
* Anular a última ação utilizando o botão de UNDO ou pressionar “Ctrl” + “Z”

Para selecionar maioria destas ferramentas, utilizam-se botões clicáveis. A palete de cores abre um pop-up, onde se pode selecionar a cor desejada e o slider de grossura é um slider interativo.

Ao selecionar uma ferramenta, o cursor muda de imagem para a ferramenta escolhida e ao mudar de cor principal, a cor deste também muda.

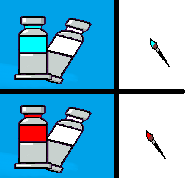


Figura 9-Mudança da cor do cursor

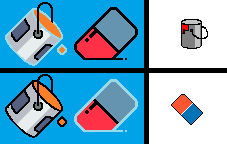


Figura 8- Mudança da forma do cursor

## 1.4 Play (Pen-Goo)

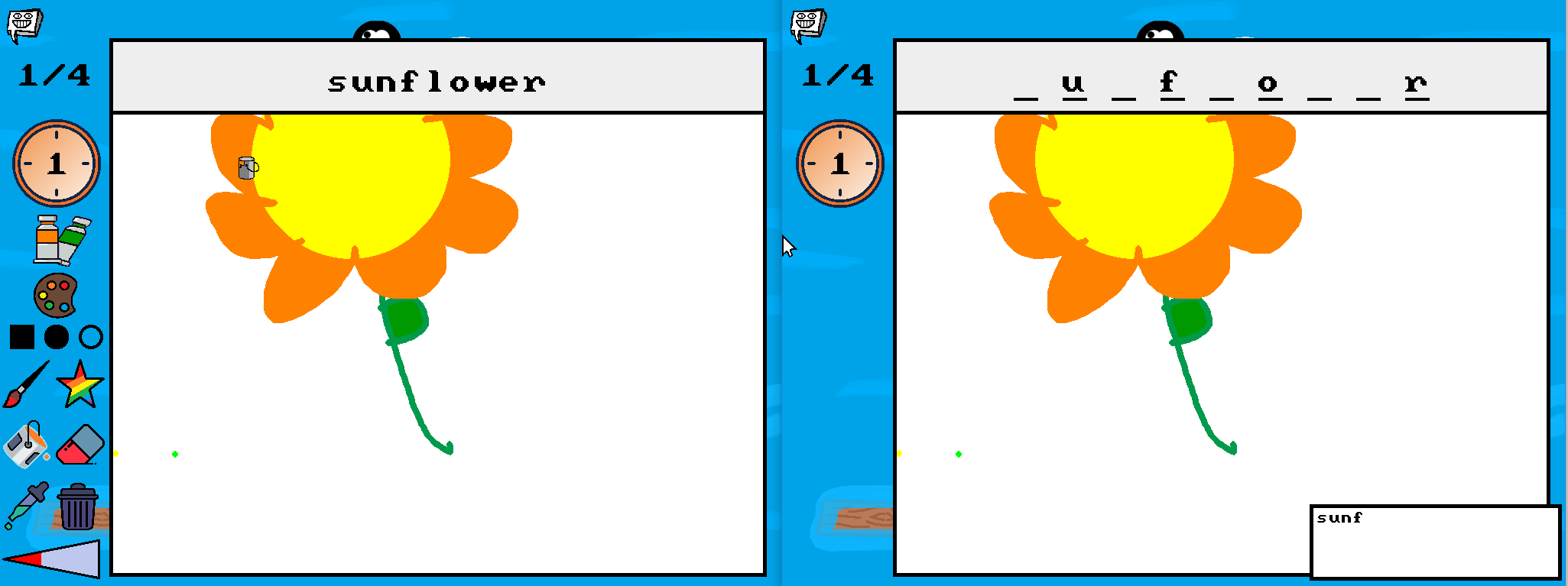
Modo principal do projeto, trata-se de um jogo de adivinhar cooperativo para dois jogadores. O objetivo do jogo é adivinhar a palavra que o parceiro está a desenhar o mais rápido possível. Cada sessão tem 4 rondas. O papel de cada jogador é escolhido quando se começa a jogar aleatóreamente.Cada uma desenvolve-se da seguinte forma: É dado ao jogador que vai desenhar uma palavra aleatória do dicionário, o jogador tenta representar o que desenha, e a mesmo tempo, noutro computador, utilizando a porta de série, o outro jogador tenta adivinhar o que o parceiro está a tentar representar , usando o teclado, o mais rapidamente possível. O desenho é transmitido em tempo real para o computador do parceiro. O jogador que desenha tem acesso a todas as ferramentas de desenho do modo “Draw”.

Figura 10 - Pen-Goo (vista dos 2 jogadores)

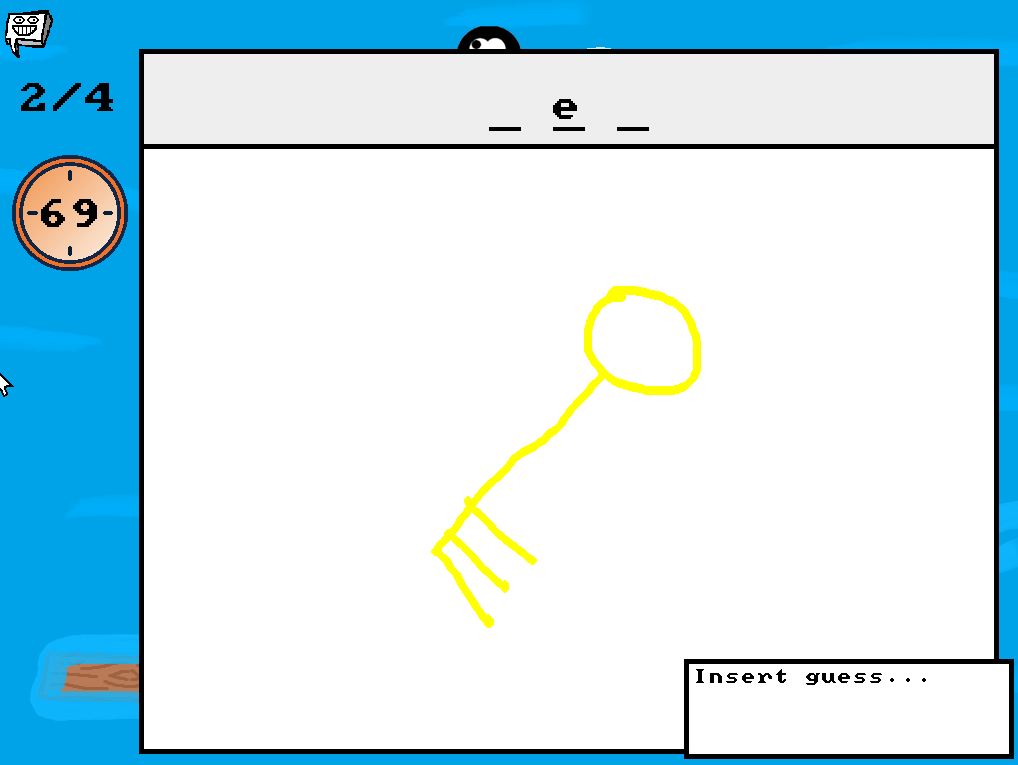


Figura 11 - Caixa de texto para tentar adivinhar a palavra

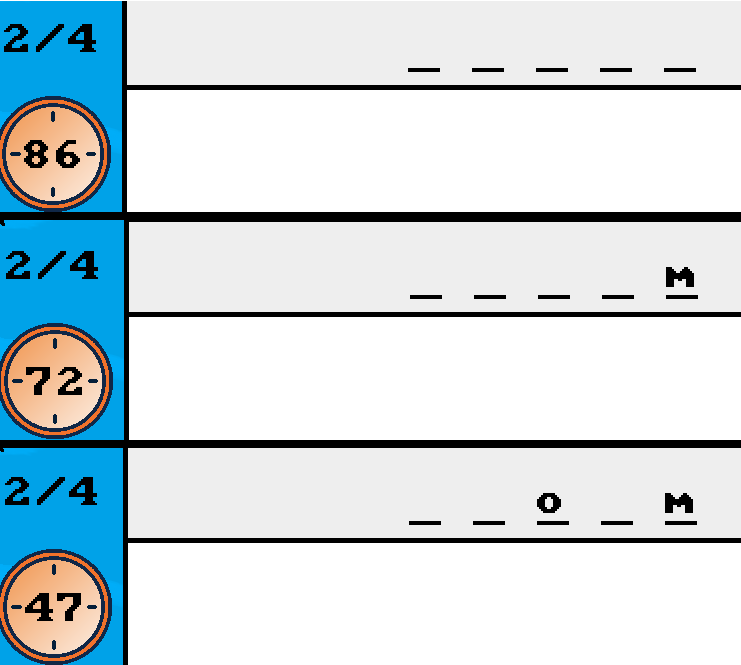
Para além do desenho do seu colega, o jogador que tenta adivinhar também saberá quantas letras tem a palavra que tenta adivinhar, e com o passar do tempo algumas delas vão sendo reveladas.

Figura 13 - Ecrã de palavra adivinhada com sucesso

Figura 12 - Letras a serem reveladas à medida que o tempo passa

Quanto mais rápido o jogador adivinhar, mais pontos o grupo ganha na ronda.

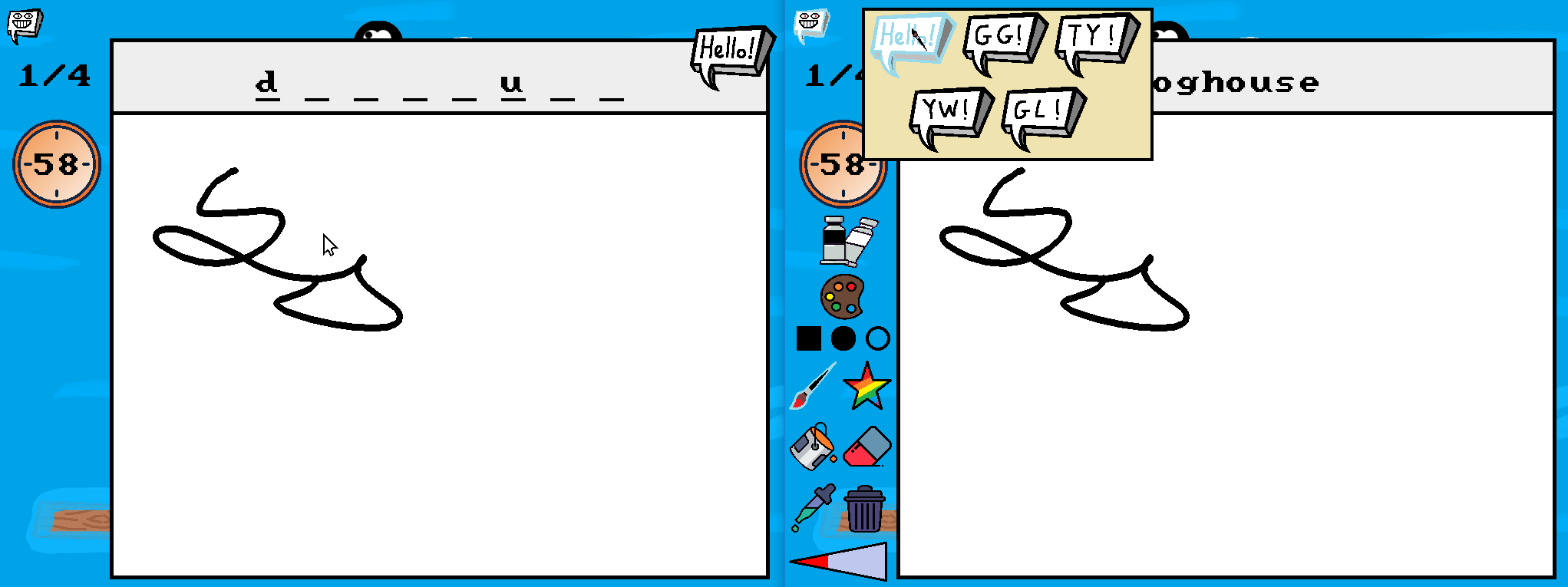
Os jogadores podem comunicar entre si utilizando emotes.

Figura 14 - Sistema de emotes (visto da prespetiva dos dois jogadores)

## 1.5 Draw Together

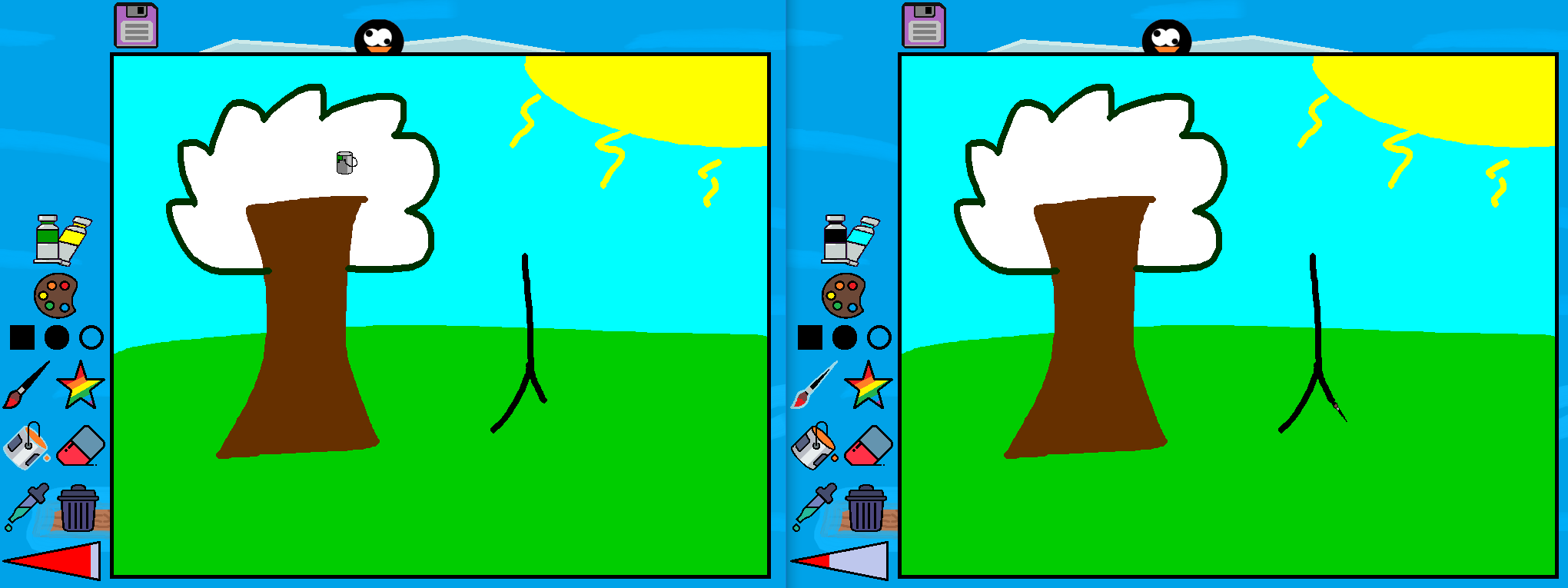
Modo de jogo igual ao “*Draw*”, mas com um parceiro, através da porta de série. Os desenhos estão sincronizados em tempo real. Todas as ferramentas de desenho estão disponiveis para ambos os utilizadores, e ainda têm a possibilidade de guardar o desenho, tal como no modo “Draw”.

Figura 1 - Modo Draw Together

## 1.6 Rainbow Snake

Figura 17-Ecrã de resultados do rainbow snake

Figura 16- Rainbow Snake

O modo de jogo *rainbow snake* trata-se de um modo de jogo secundário, que se diferencia bastante do resto do projeto. O objetivo é sobreviver o maior tempo possível, controlando a *rainbow snake*. Com o passar do tempo a *snake* vai perdendo vida e ganhando velocidade, por isso é necessário utilizar os *power-ups* disponíveis para sobreviver e maximizar a pontuação. A snake é controlada com as setas do teclado , podendo andar em qualquer direção. Atingir os obstáculos faz com que a snake perca vida e atingir uma parede faz com que seja refletida e que perca velocidade.



Figura 18 – Power-ups de rainbow snake, visualizados nas instruções do jogo

O highscore é gravado num ficheiro de texto.

## 1.7 Flappy Rainbow

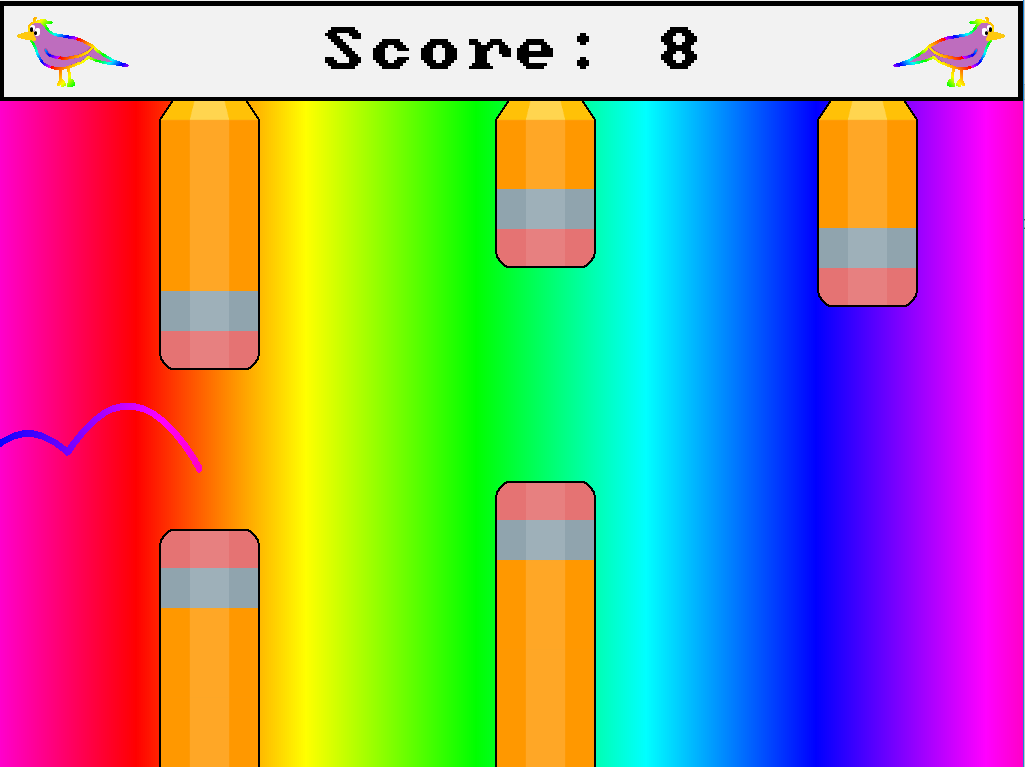
Outro modo de jogo secundário, *Flappy Rainbow*, baseado no conhecido jogo mobile, *Flappy Bird*, consiste em sobreviver o máximo de tempo possível, evitando obstáculos, apenas com salto (tecla “space bar”) e conhecimento da física do jogo. O *flappy rainbow* tem velocidade e aceleração (para baixo), para tornar o jogo mais consistente com a realidade. O jogador perde quando o *flappy rainbow* colidir com um dos obstáculos. Tal como o *rainbow snake*, o highscore é gravado num ficheiro de texto, para registo futuro.

Figura 19 - Flappy Rainbow

# 2. Estado do Projeto

|  |  |  |
| --- | --- | --- |
| Dispositivo | Utilização | Interrupcão? |
| Timer | Controlo da frame rate nos minijogos, revelar letras periodicamente, animações a mostrar scores. | Sim |
| Teclado | Escrever texto no ecrã, controlar *rainbow snake* e *flappy rainbow*, alguns *shortcuts.* | Sim |
| Rato | Navegação nos menus, desenhar, escolha de ferramentas de desenho. | Sim |
| Placa Gráfica | Visualizacão de imagens, desenho e menus. | Não |
| RTC | Mostrar data e hora, atualização do relógio de jogo. | Sim |
| Porta de série | Modos de jogo multiplayer “Play” e “Draw Together” | Sim |

// é preciso dizer se e por polling ou interrupts , qual a funcao dele , como foi implementado e em que ficheiro foi implementado.

* **Timer**: O timer 0 é usado para controlar a framerate, atualizar o desenvolvimento do jogo nos modos “*Rainbow Snake*” (*snake.c*) e “*Flappy Rainbow*” (*flappy.c*), para controlar o tempo entre letras reveladas (*wordpicker.c – reveal\_letter()*), e para mostrar a pontuação final não espontaneamente.

Está implementado em: timer.c

* **Teclado:** O teclado é utilizado para escrever no ecrã , a partir de caixas de texto (*textbox.c*) (adivinhar palavra, guardar nome de ficheiro, fazer load a ficheiro) , para sair de certos modos de jogo e fazer *undo* no desenho (“Ctrl” + ”Z”) , mas também para controlar a *snake* (setas) (*snake.c – snake\_turn\_left/right()*) e o flappy (“Space bar”) (*flappy.c -flappy\_jump()*).

Está implementado em : keyboard.c

//acho qeu nao e isto que eles querem com os ficheiro / funcoes.... problema deles

* **Rato:** Usado para controlar o cursor, que é utilizado para navegar nos menus e selecionar ferramentas, a partir de botões e desenhar, não só para escolher onde desenhar mas também para escolher se se utiliza a cor principal (botão esquerdo do rato) ou secundária (botão direito do rato).

Está implementado em : mouse.c

* **Placa Gráfica:** Este dispositivo é utilizado para mostrar todas as imagens usadas no programa (bitmap , RGB(8;8;8)). O modo de vídeo utilizado é 118 (1,024×768 (24 bits p pixel)). (*proj.c*)

Double buffering está implementado no código do programa, mas não é utilizado, pois devido ao facto de que se trata de uma ferramenta de desenho, não muito focada em atualização frequente do ecrã inteiro, não faria muita diferença, tornando a experiência mais lenta até. (*video.h*)

Foi utilizada colisão de *sprites* para o cursor e botões, e hitboxes para os modos de jogo “Rainbow Snake” e “Flappy Rainbow”. (*sprite.c - is\_on\_button()*) (*sprite.c- check\_hitbox\_collision()*)Um sistema de *layers* está a ser utilizado de modo a ser possível desenhar facilmente janelas por cima de outras sem perder a informação que está “por baixo”. (*layer.c*)

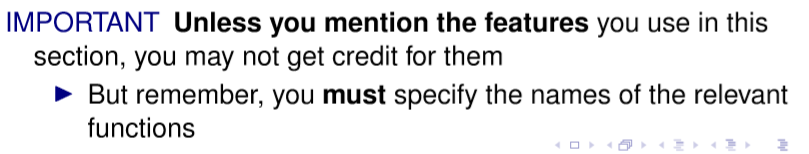
A fonte utilizada para a textbox, para mostrar as pontuações dos jogadores, tempo do relógio, etc., é retirada de um bitmap que contém todos os carateres, que depois são retiradas individualmente a partir de um algoritmo. (*textbox.c*)

Está implementado em: video.c vbe.c

* **RTC :** O real time clock utiliza alarmes de 1 segundo para atualizar a hora / data atual periodicamente no Ecrã inicial (*clock.c – update\_big\_ben()*) e para atualizar o relógio de jogo (90s por ronda no máximo). (*wordpicker.c – wordgame\_tick\_clock()*)

Está implementado em : rtc.c

* **Porta de série:** Mt importante,...



# 3. Organição / Estrutura de código

Mouse

-Codigo importado do lab4

-Utiliza-se para subscrever as interrupcoes do rato e interrupt handler rato.

Rtc

- Le a informacao dos registos do rtc

- subscreve alarmes periodicos de 1 segundo , para ler a data/horas em todos os segundos.

-Uso das struct Date / Mini\_Date para obter as datas a partir das funcoes rtc\_get\_date/mini\_date();

Timer

-Codigo importado do lab2

-Utiliza-se para subscrever as interrupcoes do rato e interrupt handler timer.

Keyboard

-Codigo importado do lab3

-Utiliza-se para subscrever as interrupcoes do rato e interrupt handler kb.

Vbe

Video

Uart

Uart\_protocol

Uart\_wordgame

Interrupts

-funcoes para dar subscribe e unsubribe e receber notificacoes de todos os perifericos utilizados de forma mais limpa e intuitiva.

-Uso da struct Notification para , apartir da funcao GetNotification() obter todas as informacoes convenientes da notificacao recebida.

-

PenGoo

- Base do projeto

- Lógica principal de todos os menus e dos modos pricipais

-

-

-

-

Bitmap

-Funcoes para Fazer load / save a bitmaps

-Struct utilizadas !!

-Transparecia ?

Bitmaps

-Le todos os bitmaps utilizados

Sprite //meter funcoes importantes

-sprite , hitbox , button , idle sprite , slider

-sprite : draw (tem em atencao o rosa – invisivel , e o verde – cor mudavel), erase , move

-cursor : update , mudar de bmp , mudar de cor

-hitbox:

-button: draw button(s) , highlight ,(un)press

-iddle sprite:

-slider:

Layer

-sistema de layers que permite desenhar sprites numa layer em cima de outra layer , sem perder informacao sobre o que esta por baixo

Is\_top\_layer(), draw\_on\_layer() ,

Canvas

-Canvas utilizado para desenhar.

-struct canvas .(com limites)

-Funcoes de desenho canvas\_draw\_line(), ...

-Ferramentas de desenho – bucket\_tool , rainbow

-Shapes

-Undo ( canvas\_save\_drawing(), canvas\_undo())

Textbox

-Utilizado para escrever texto

-bitmap da fonte ( como e que se vai buscar cada letra + funcao)

-cursor da caixa de texto

-Funcoes?

Clock

-Relogio utilizado para mostrar as horas

-Utiliza rtc

Emote

-Sistema de emotes usados em multiplayer entre jogadores

Wordpicker

-Mecanismo de mostrar a palavra que esta a ser adivinhada

-Ler o dicionario de palavras

-Mostrar palavra escondida wordgame\_draw\_hidden\_word() , erevelar letras reveal\_letter() , e verificar a tentativa de adivinhar verify\_guess()

Snake

-Minijogo de snake

-Struct snake - parametros

-Colisoes

-Fisicas da snek snake\_move() , snake\_turn() ;

-Struct arena

-Power ups – Consume , arena\_add\_powerup()

-HP bar

-Stats

-Load / save highscore

Flappy

-Minijogo flappy

-struct Bird – parametros – aceleracao e velocidade !!!

-Jump , colisoes

-Tubos

-Stats

-Load / save highscore

Game\_info

-Instrucoes de cada modo

-Acedidadas a partir do menu principal

Game\_State

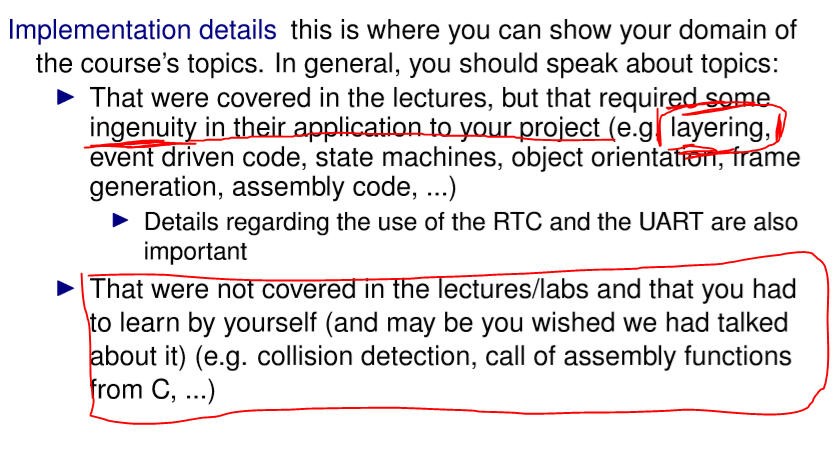
!!!!!

|  |  |  |
| --- | --- | --- |
| Modulo | Peso relativo(%) | Feito por |
| Mouse |  |  |
| RTC |  |  |
| Timer |  |  |
| Keyboard |  |  |
| VBE |  |  |
| Video |  |  |
| Uart |  |  |
| Uart\_Protocol |  |  |
| Uart\_Wordgame |  |  |
| Interrupts |  |  |
| PenGoo |  |  |
| Bitmap |  |  |
| Bitmaps |  |  |
| Canvas |  |  |
| Sprite |  |  |
| Textbox |  |  |
| Layer |  |  |
| Canvas |  |  |
| Clock |  |  |
| Emote |  |  |
| Wordpicker |  |  |
| Snake |  |  |
| Flappy |  |  |
| Game\_info |  |  |
| Game\_state |  |  |

## Call Graph

**INSERIR GRAFICO AQUI**

# 4.Detalhes de implementacão



-Detalhes de uart e rtc

-Layers

-algoritmos mais dificies ( de linhas , cicrulos , balde ) deve haver coisas boa para aqui

-Bitmaps

# 5.Conclusao

-Lcom e pessimo

-No inicio ninguem sabia o que se passava

-Agora maioria das pessoas nao sabe o que se passa

-Mas o projeto ate e fixe

-sys\_inb é gay